(2)

AFHRL-TR-90-9

AIR FORCE

HUMAN RESOURCES

AD-A223 839

METHODOLOGY FOR EVALUATION OF AUTOMATION IMPACTS ON TACTICAL COMMAND AND CONTROL (C^2) SYSTEMS: IMPLEMENTATION

Kevin M. Corker
Nichael L. Cramer

BBN Systems and Technologies Corporation
10 Moulton Street
Cambridge, Massachusetts 02238

Eugene H. Henry, Captain, USAF

LOGISTICS AND HUMAN FACTORS DIVISION
Wright-Patterson Air Force Base, Ohio 45433-6503

June 1990

Interim Technical Report for Period February 1988 - January 1990

LABORATORY

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235-5601

NOTICE

This report has been reviewed and is approved for publication.

BERTRAM W. CREAM, Technical Director
Logistics and Human Factors Division

HAROLD G. JENSEN, Colonel, USAF
Commander

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
June 1990

3. REPORT TYPE AND DATES COVERED
Interim - February 1988 - January 1990

4. TITLE AND SUBTITLE
Methodology for Evaluation of Automation Impacts on Tactical Command and Control (C^2) Systems: Implementation

5. FUNDING NUMBERS
C - F33615-87-C-0007
PE - 62205F
PR - 3017
TA - 08
WU - 06

6. AUTHOR(S)
Kevin M. Corker
Nichael L. Cramer
Eugene H. Henry

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
BBN Systems and Technologies Corporation
10 Moulton Street
Cambridge, Massachusetts 02238

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Logistics and Human Factors Division
Air Force Human Resources Laboratory
Wright-Patterson Air Force Base, Ohio 45433-6503

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFHRL-TR-90-9

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This project is developing a methodology for evaluating the impact of automation on tactical command and control (C^2) operators. The focus is on prototypic man/machine systems and their automation alternatives in a simulated system context. Although the methodology has initially been developed to simulate operators using a soon-to-be-fielded tactical ground control radar system, the methods, models, and software tools are generally applicable for use in other cases. A modular architecture promotes this feature, as well as experimentation with different modeling approaches.

The methodology, hosted on a mini-computer workstation, is composed of three major components: human performance models, which consist of cognitive, visual, auditory, and psychomotor components; rapid prototyping tools for prototyping of man/machine systems and interfaces; and utilities for specifying a scenario with which to exercise the system and human models. The designer/analyst can modify the configuration and function of the system or the assumptions underlying human performance through screen-based utilities. A simulation of a scenario produces human performance data as if the scenario were executed by real operators. The workstation provides a timeline of events, activity profiles, and subjective workload estimates.

The present report documents the implementation of this evaluation system.

14. SUBJECT TERMS

automation	object-oriented programming
C^2 systems	simulation
human performance	work-load evaluation

15. NUMBER OF PAGES
44

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

SUMMARY

This report documents the implementation of a software system designed to support a methodology for analytic evaluation of the impact of automation on the performance of US Air Force command and control (C^2) systems. The evaluation is performed using a set of software tools to emulate the operation of ground control radar systems, and to examine the effect of automated aids applied to tactical C^2 procedures. The tools allow an analyst to set up a tactical C^2 scenario, to define the operational characteristics of the ground control radar consoles, and to specify the human performance characteristics of the system operators. The analyst can then run the simulation, observe the actions and procedures of the simulated operators, and receive performance and workload measures associated with the configuration to be tested.

The particular C^2 operation that is the focus of this effort is the Control and Reporting Center (CRC) operation, an element of the USAF Tactical Air Control System (TACS). CRC operations have recently been the focus of automation upgrades. Litton Industries, Inc. has provided a semi-automated system, called Modular Control Equipment, to perform CRC operations. The Modular Control Equipment (MCE) CRC operation is serving as the test bed for this evaluation methodology. A more extensive description of the operation of the CRC and the rationale for its selection to exercise this methodology is found in AFHRL-TR-89-17 (Methodology for Evaluation of Automation Impacts on Tactical Command and Control (C^2) Systems: Domain Selection and Approach).

The present report describes the software implementation of this evaluation methodology. Background concerning the TACS mission and CRC/MCE operation is briefly provided to establish the context for implementation discussions. This background includes a discussion of the requirements for analytic simulation in prototype automation equipment development. A description of our object-oriented simulation paradigm is then provided. We then discuss the software architecture specific to this evaluation methodology, including descriptions of the equipment representation, the basis of the performance model for the human operators, the scenario functions, and the utility of the simulation output in assessing the impact of automation on C^2 systems.

PREFACE

The implementation of the C^2 evaluation methodology to assess the impact of automation on the performance of C^2 systems is described in this document. This is a second Interim Report under USAF Contract #F33615-87-C-0007. The first Interim Report (AFHRL-TR-89-17) provided information about the context of and requirements for an efficient and effective evaluation methodology to be applied to emerging automation initiatives in the area of tactical C^2. This document describes the software implementation of the evaluation methodology and its operation. The work described is the basis for an ongoing development effort that will include use of the software simulation to investigate human operation of advanced automation in tactical C^2 systems. Provision for hybrid simulation and human interaction with the evaluation software, as well as linking that software to other USAF C^2 systems, is discussed.

This work could not have been performed without the extensive assistance of the staff of the Air Force Human Resources Laboratory (AFSC) at Wright-Patterson Air Force Base. The Air Force program manager was Captain Eugene Henry. In addition, Major Donald Smoot provided operational expertise concerning command and control. Their cooperation and guidance, as well as broad subject-matter expertise, have contributed significantly to the project.

TABLE OF CONTENTS

List of Figures

List of Tables

Methodology for Evaluation of the Impact of Automation on Tactical Command and Control (C^2) Systems: Implementation

I. INTRODUCTION

This report describes the development and implementation of a methodology to evaluate the impact of automation on United States Air Force (USAF) command and control (C^2) systems. This methodology is intended to provide efficient and effective evaluation of the operational impact of the automation initiatives being introduced into the tactical C^2 environment.

Recent developments in C^2 systems are influenced by several trends: (a) the need to make quicker decisions; (b) the need to process increasingly large streams of data; and (c) the need to make current large, relatively immobile C^2 facilities less vulnerable. These trends, combined with the recent availabilities of small and powerful computer systems, have allowed the development of highly automated C^2 facilities. The impact of these automation initiatives will be a far-reaching change in the way C^2 operations are performed. To account for these changes, the Air Force will have to re-examine its tactical and strategic C^2 planning across echelons. In addition to having an impact on tactical operations, the introduction of automation into C^2 systems will fundamentally affect the current human/machine interaction process. This effect, the operational and procedural impact of automation on man/machine interaction, is the current focus of our evaluation methodology.

It is critical that the man/machine interaction be studied early in the design process. Design for operability, appropriate information presentation, handoff between man and machine, procedural coordination, and communication requirements must be an integral part of system development. Concentration on the development of automated capabilities without attending to full man/machine integration can seriously degrade system operation, with costly consequences. Once design decisions have been made, it is difficult to have them amended. Further, retrofit, if possible, is vastly more expensive than original design; and retrofit designs, facing a set of full system constraints, often fail to reach the level of performance attainable by an early design for usability (Rouse & Boff, 1987). Our evaluation

methodology, therefore, explicitly addresses human performance and human interface issues. Design for performance evaluation in prototype system development imposes the requirement to provide a rapid development test-bed system, the characteristics of which can be manipulated to accommodate progressive design changes in equipment. The approach we have developed and implemented meets this requirement by providing a software-based workstation, a system of models, and a set of tools to aid the designer/analyst of advanced Air Force C^2 systems. The purpose of this workstation is to anticipate the effect of the introduction of automation into complex C^2 systems and to provide predictive measures of human performance in those systems.

To meet this purpose, the methodology must satisfy two major requirements. First, it must simulate human performance and human interface with automated equipment. Second, it must provide assessment utilities that predict C^2 system performance levels across a range of non-automated, partially automated, and fully automated tasking in the C^2 environment. The first of these requirements is met through the implementation of a set of human performance models. The second requirement is met through provision of facilities to simulate and manipulate the tactical C^2 environment.

The next section describes the C^2 tactical environment. Section III presents the implementation of models and scenarios. Section IV describes the system architecture, and Section V presents our conclusions.

II. C^2 TACTICAL ENVIRONMENT

The USAF Tactical Air Control System (TACS) is responsible for the planning and execution of tactical (i.e., within an operational theater) air-to-air and air-to-ground operations. It consists of several echelons and elements. At the upper level is the Tactical Air Control Center (TACC), which is responsible for overall battle planning (issued in the form of Air Tasking Orders or ATOs) and for conduct of the deep strike missions such as Air Interdiction (AI) and Offensive Counter Air (OCA). Subordinate to the TACC, and responsible to it for the conduct of the Defensive Counter Air (DCA) or air defense mission, are several echelons of radar-based elements: the Control and Reporting Center (CRC), the Control and Reporting Post (CRP), and the Forward Air Control Post (FACP). For the conduct of Close Air Support (CAS) and Battlefield Air Interdiction (BAI) missions for air-

to-ground operations at the forward edge of the battle-area (FEBA), the Air Support Operations Center (ASOC) and the currently-under-development Ground Attack Control Capability (GACC) are also subordinate to the TACC.

In developing a methodology for evaluating the impact of automation on tactical C^2 operators, the major characteristics of the tactical operations must be considered: Decision-making, chain of command, communication exchange, and selection of courses of action are the core of tactical C^2. These functions are highly affected by the goal states of the C^2 element and individual decision-maker within the context of the tactical situation at the time of the decision. Tactical C^2 operations are highly procedural, but the selection of appropriate procedures is very situation/context-sensitive. Some procedural decisions are based on semi-rigorous assessment of the situation; others are almost purely heuristic and based on recognition of a pattern or situation.

Tactical C^2 has a strong hierarchical aspect. There is a hierarchy among the C^2 elements and echelons, among the operators within a given C^2 element, and among the goals and activities of a given operator. These hierarchies seldom mix in a completely straightforward manner. It is entirely possible for some actions (such as attending to an aircraft emergency) that are initiated by a Weapons Director (WD) at a CRC to take precedence over those directed by the WD's supervisor, the Weapons Assignment Officer (WAO). This mixing of goal, structural, and activity hierarchies also tends to promote frequent inter-activity interruption. In some cases the original activity is resumed at the conclusion of the interruption, in some cases it is restarted, and in some cases it is forgotten. Although there is hierarchical influence on the decision-making (personnel usually carry out their supervisor's directives and try to achieve the goals they have been given), much of each individual's decision-making derives from a common understanding of the situation. This situation assessment process relies heavily on inter-operator communication.

A significant proportion of the communications activity focuses on exchange of information about the context and situation to optimize this distributed individual decision-making. This situation assessment and information-sharing are of critical importance to the operators, and have some interesting characteristics. First, often a substantial amount of information will be exchanged without producing any observable result. The operators are either adding to or confirming their internal representations of the tactical scenario. Second,

this information exchange is likely to be very sensitive to increased facility modularization and physical dispersal.

Typical activities supported by the hierarchy and communications described above include manning Combat Air Patrols (CAPs), scrambling friendly aircraft, and pairing. Manning the CAPs is simply a matter of ensuring that a set number of friendly fighters are or will be orbiting a navigation point (i.e., CAP). Scrambling is the request to an airbase to have more aircraft become airborne to man CAPs or to engage hostile aircraft. Pairing is the assignment of friendly fighter aircraft to a hostile aircraft for the purpose of visual inspection, escort, or attack.

Effect of Automation on TACS

With the exception of limited computer capability in the CRC/CRP's current 407L system and some automation of the ATO generation process at the TACC, virtually all TACS elements have been predominantly non-automated. However, by the mid- to late-1990s, the TACC Modernization Project will introduce widespread upgrades and modernization that will affect virtually all TACS functions and elements. Specifically, the TACC Modernization Project will provide improved overall force planning and management. Furthermore, the fielding of Modular Control Equipment (MCE) in the CRCs, CRPs, and FACPs will greatly expand the capability of these radar-based elements through automation. Finally, the development of the GACC (as a derivation of MCE) and upgrade to the ASOCs will add new capability and speed to the management of air-to-ground operations.

Common to all of the TACS system upgrades and replacements are a large increase in system automation, a modularization and physical dispersal of the facilities containing the operators and equipment, and a distribution of functions among the various modules of any given TACS element. Although any one of these changes, such as the dramatic increase in system automation, would be significant, the combination of <u>all</u> would be better characterized as revolutionary instead of evolutionary in effect. The following comparison with the current environment reveals some of the effects of upgrading and modernizing the TACS.

The introduction of automation into the TACS promises great increases in ground control capability, improvement in mobility and modularity, and a decrease in the number of

personnel required in a given area of responsibility. There are, however, a number of issues that attend the introduction of automated systems in TACS.

1. Will human workload saturate given a particular system and are procedural bottlenecks revealed? Given an ability to handle several times the number of radar tracks that current ground control systems can handle, when do the human operators begin to reach performance limits and how will they shed excessive load?

2. What will the duty cycle or workload of an operator be in an automated system? What are the transient or peak loads that can be handled and what are the long-term strains that will be encountered? What are the procedural differences between current and advanced systems? On a broader scale, what are the tactical and doctrinal differences that the advanced control capability will impose on current TACS operational standards?

3. What is the impact of automation initiatives on manpower and training for new systems? Given the complexity of rapidly reconfigurable software and firmware-based control systems, what are likely sources of operator error, and what demands for special training will be incurred? A system could be rendered ineffective if operators are not able to exploit the full range of system features because of inadequate training.

4. What is the effect of automation on the information and data requirements for system operation? In designing for optimum information flow, the designer must determine the paths and media through which to supply information. Automation provides the C^2 designer with flexibility, but raises new issues as to the form (semantics) and method (syntax) of providing operator data. The inevitable increase in data that automation provides must be balanced by design to avoid operator overload.

5. How can automation be effectively transferred into the TACS elements? How will personnel who are experienced with existing systems adapt to the new automated facilities and procedures? How will automated operations be integrated with non-automated systems? How will the system transition be implemented?

6. What are the procedures associated with system verification and validation? The introduction of automation raises new challenges for operational, integrated operator/system testing.

The success of the introduction of automation into a complex network of responsibility such as TACS depends on timely answers to these questions. However, traditionally it has been difficult to predict the impact of prototype and developing systems prior to fielding and testing. The current effort attempts to address that difficulty with a predictive evaluation simulation methodology to determine the impact of automation in relation to the issues mentioned above.

Effect of Functionally Distributed, Physically Dispersed TACS

The MCE-equipped CRC introduces yet another revolutionary change in TACS operation. The MCE modules (each supporting four Operator Control Stations, or OCSs) are to be manned by a mix of identification, weapons control, and battle management personnel. These operators have differing responsibilities in C^2 operations. Common situation awareness will need to be maintained among crewmembers through the operation of the OCS and through voice channel communication. The impact of these operational constraints on the MCE is not clear. On the one hand, the MCE system allows operators to share a common representation through the Radar Graphics Display Unit (RGDU) in the OCS. On the other hand, the system imposes physical separation on operators and forces a reliance on voice intercom for most information exchange. Our evaluation methodology is designed to be sensitive to verbal communication protocols. In particular, we seek to identify communication "bottlenecks" in which the procedures associated with intercom use cause task interruption or message confusion.

Effect of TACS Operational Environment on Evaluation Methodology

Abstracting a set of operational characteristics from the above discussion, we find that the TACS environment provides the following challenges to the development of an evaluation simulation methodology.

First, the simulation must be able to represent multiple independent agents, with hierarchies of goals and activities. These agents must be able to respond to a representation of C^2 operational requirements according to individually tailored responsibilities and rules.

Second, the evaluation methodology must provide a mechanism and structure for precedence and priority in agents' actions. Because of the military command structure, some agents' authority exceeds that of others; and because of the nature of C^2 operations, some actions and procedures are more important than others. The methodology must provide decision-making procedures that are dynamic and sensitive to the nature of the operations. Ideally, the priority structure should be dynamically reconfigurable.

Third, a mechanism to represent interruption must be incorporated. The methodology should account for the interruption process; predict its effect on performance; and direct the methods whereby activities are resumed, restarted, or aborted.

Fourth, the methodology must adequately describe inter-operator communication. This description should specify communication protocols, as well as the effect of communication, in that communication among operators provides the means by which individual world knowledge is shared in a common representation with other crewmembers. Further, communication between the ground crew and pilots is the basis of response to ongoing operational events.

With these challenges in mind, the goals of the implementation of this simulation methodology are as follows:

1. To refine and apply knowledge acquisition and representation techniques in order to provide functional analysis and simulation of a human/machine system.

2. To tailor and apply human/machine performance models to mimic full system functionality.

3. To develop an effective user/analyst interface for interaction with the simulation.

4. To provide tools to exercise and evaluate the insertion of automated aiding functions in the system through adjustment of simulation parameters and models.

III. IMPLEMENTATION

This section describes the implementation of the C^2 evaluation methodology and discusses the functional requirements for modeling the system and its human operators. Section IV will discuss the architectural details of the computer code that supports these functions, and the data control flow through that architecture.

MCE Functional Description

As mentioned previously, the MCE-equipped CRC is the test case to which this methodology was applied. Now, we briefly describe the specific functional characteristics of this system as an introduction to its implementation in our simulation evaluation.

A CRC has four general functional responsibilities:

1. Overall air defense battle management;

2. Detection and tracking of all aircraft within its area of responsibility (AOR);

3. Identification of all tracked aircraft; and

4. Weapons allocation/control of fighters to visually identify (VID) unknown aircraft (those that cannot be identified by other methods) and to intercept or shoot down those identified as hostile.

These responsibilities are identical for both current CRC equipment installations (407L) and MCE-equipped operations. The methods and manning by which these responsibilities are met differ radically between the two systems, however. Table 1 (MCE vs 407L) illustrates these differences.

Table 1. Operating Characteristics: MCE versus 407L. The duties and manning associated with 407L are derived from observation of that system operation. The manning and function for the MCE come from observation of that system's operation at the US Marine Corps Base at Camp Pendleton, CA, and from discussions with MCE instructors at Luke AFB, AZ.

FUNCTION	407L		*1 OPERATIONS MODULE* Modular Control Equipment (MCE)	
	No. of Operators	Level of Automation	No. of Operators	Level of Automation
Battle Management	Manager Technician	Low	Battle Director	Low
Weapons Allocation	WAO Technician	Moderate	WAO	Moderate
Weapons Direction	Weapons Controllers Technicians	Low	Weapons Directors	High
Surveillance/ Identification	Surveillance Operators Technicians	Low	Surveillance Supervisor	High
Equipment Support	Engineering/Computer Operators	Low	Modular Equipment operation, line replacement design	High

The table illustrates the trend to reduce the number of operators and supplement that functionality with automation as one transitions from 407L to the MCE. It is interesting to note that overall battle management and WAO functions are not significantly changed between the two systems. These high-level decision-making, planning, and logistics functions remain dependent on the human operator's cognitive abilities. The processes of identification and direct weapons control have, however, been subject to increased automation. One might speculate that automating the lower-level ground-controlled intercept (GCI) processes may have an impact on the workload and pace associated with the higher-level battle management functions. It is, in part, the purpose of this methodology to support the system designer/analyst in investigating such hypotheses.

In addition to the manning/procedure differences, the physical layout of the MCE-equipped CRC has a significant impact on operations. This impact must be captured in our simulation implementation.

An MCE-equipped CRC will have one or more (nominally four) Operations Modules (OMs) (also called Operation Control Modules or OCMs) dispersed up to 500 meters apart and connected by various data and voice communications circuits. Each OM is an 8-ft x 16-ft x 8-ft compartment containing a set of computers, radios, and four operator console units (OCUs).

Each OCU is equipped with two cathode-ray tube (CRT) displays, a voice communications access unit (VCAU) panel (to control intercom, telephone, and radio communications), and a keyboard for data entry. The primary CRT display is a 19-in. x 25-in. color unit known as the Radar Graphics Display Unit (RGDU) for display of radar and other situational data. The secondary CRT, a 17-in. monochrome unit known as the Auxiliary Display Unit (ADU), presents the operator with virtual switch panels, data entry menus, and alphanumeric displays of track data. Both CRTs have touch-sensitive surfaces to allow interaction without the use of pointing devices (e.g., a trackball or mouse). The primary interaction mode is through the use of a Finger-on-Glass (FOG) technique.

Implementation of Methodology
Functional Description

The C^2 evaluation methodology has been implemented to operate in two modes. The first mode of operation is that which provides analytic and predictive performance data based on human and system simulation models. We have termed this the "analytic mode." The second mode is that which supports operation of the C^2 evaluation workstation in a hybrid mode of simulation, with concurrent operation by software representation of the MCE crew and a human operator. We have termed this the "manned-simulation mode." The system has been implemented in a modular fashion such that, in addition to these standard operational modes, external inputs and subsystems can be easily accommodated to support integration with other external independent systems that are part of the USAF C^2 operation.

Analytic Mode Operation

In the analytic mode, the system is driven by a simulation scenario. The response of the operators of the MCE to the analyst-specified scenario is generated by models of human performance that are tailored to individual operator responsibilities. These models respond to

the stimuli that are presented to them via emulation of the MCE OCM. The human operator models process incoming information, interact with the simulated MCE equipment, and communicate with each other via simulated message traffic. Performance analysis is provided through operation and interaction of the human operator models with the OCM emulation. These models provide prediction of individual operator response to the simulation scenario in terms of actions, perceptual response times, and performance accuracy. Execution of the selected action is modeled through motor response times and accuracies. The effect of operator action is "displayed" through appropriate response of the MCE equipment simulation. The analytic mode of operation also generates functional interactions among operators and mimics the procedural requirements for communication and data exchange. Each operator's rules of behavior are modeled according to his/her function in the MCE. This modeling includes duty assignments and interaction protocols among the operators. The individuals or agencies modeled, and their responsibilities, are described in Table 1 (MCE vs 407L).

The rationale for the selection of the types of models that are used to describe the human response to MCE operations is provided in detail in Corker, Cramer, Henry, and Spaeth (1989). The modeling architecture and the models themselves will be described in detail here to provide a context for the forthcoming discussion of the software structure that implements those models.

Manned-Simulation Mode

In this mode of operation, the C^2 workstation, through its emulation of the MCE equipment, can be used to provide input from an actual human operator to the simulation scenario and to support interaction of that operator with the simulated operator objects. This operational mode is supported by the current software implementation, though the hardware required to fully exercise this functionality has yet to be acquired and integrated into the system. The design for this integration includes a voice recognition system to interpret the human operator's commands, a speech generation system to provide auditory input from other MCE operator objects, and a touch panel overlay to emulate the operation of the MCE control and radar graphics units.

The current functionality provides for switch actions to be taken by a human operating as a WAO or WD. These switch actions are initiated by cursor movement to, and

selection of, the desired switch or RGDU object controlled by a mouse. The result of the switch action or radar object selection is displayed to the operator. The software architecture (discussed in detail in the next section) to support incorporation of that action into the ongoing simulation is implemented in the current system. Figure 1, human operator interaction with the simulation, illustrates the control flow that is supported by the current implementation.

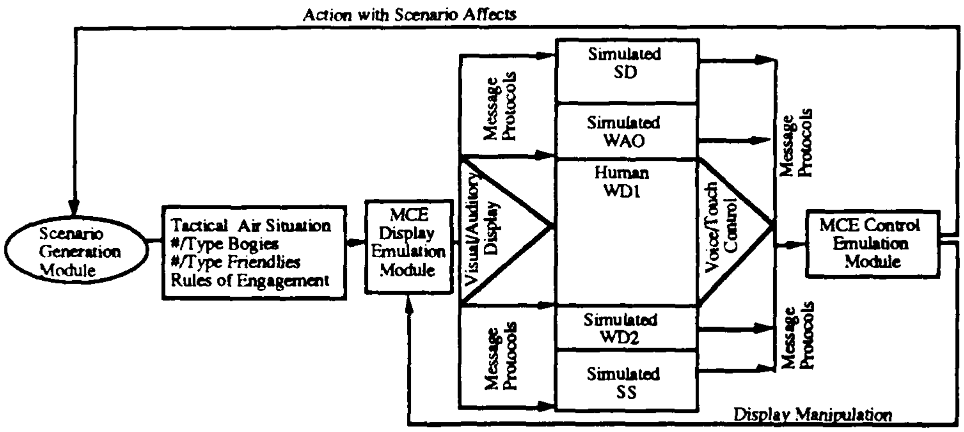

Figure 1. Human Operator Interaction with the MCE Simulation in a Manned-Simulation Mode.

Models

In developing a model-based representation of human performance in a domain as complex as tactical C^2 operations, rigorous requirements must be met in the selection and integration of those human performance models. We will now describe our model selection process and the functions we have attempted to emulate.

Model Perspective

Models in the C^2 evaluation methodology are used to describe and predict human operator response to operationally driven presentations of a tactical air defense scenario. Our general view in model selection is that of an information processing and cognitive process perspective. This perspective asserts that:

> ... human performance varies because of differences in the knowledge that a person or team of people possess (both the form and the content), in the activation of that knowledge, and in the expression or use of that knowledge. Woods, Roth, Hanes, & Embrey, 1986, p. 6.

This perspective (one of many useful views of human performance) was selected to respond to the operational and analytic requirements detailed in Section II. In order to meet the specific needs of MCE operation, model requirements are as follows:

1. We must represent human visual and auditory perceptual processing -- these being the primary modes for information exchange in C^2 operations.

2. We must provide models of cognitive processes, including memory, decision, pattern or situation assessment, processing-resource allocation, and a mechanism to guide the focus of operator attention as a function of perceptual processing, memory and plans.

3. We must provide motor and verbal response models to describe the performance of operator actions.

Even within this particular human modeling perspective, models differ in many dimensions. Although the topic is outside this discussion, we note that differences in modeling approaches are hotly contested, reflecting the complexity of the task of human performance modeling, and the lack of definitive empirical and validation for a given set of models (e.g., Elkind, Card, Hochberg, & Huey, 1989; Salvendy, 1988; Woods et al., 1986).

However, two dimensions of these controversies have direct impact on our methodology. These are the resolution of model-based description of human performance, and the mathematical assumptions that form the bases of model operation.

Resolution, or level of detail, is a modeling characteristic that deals with the issue of how much detail is necessary to provide a sufficient description of the behaviors of interest. The issue of resolution can be described in two dimensions. There is a breadth-versus-depth

aspect for human performance models. On the one hand, research models (intended to explore the basis for and mechanisms of cognitive, motor, or perceptual processes) tend to be narrowly defined and limited in generalizability. Broader operational models, on the other hand, though more applicable to the complexities of "real-world" operation, tend to be less specific and less quantitatively predictive.

The mathematical assumptions under which models are developed determine the predictive power and applicability of those models in a given domain. Static descriptive or normative models are sufficient to describe instantaneous operator behavior. The inclusion of feedback of the effects of model activity -- and, more critically, the inclusion of human operators into the evaluation simulation -- raises issues of temporal resolution, real-time response, and system stability. There is also a probabilistic versus deterministic tension in model formulation. Perceptual/motor models may be best described by relating the signal/noise distribution characteristics of stimuli to the filter and plant characteristics of the human operator. Cognitive processes such as situation assessment may be deterministically represented as rule-based, or described probabilistically using Bayesian or evidential reasoning techniques. Memory processes can, similarly, be described in terms of probabilities of recall, or deterministically described using queuing theory techniques.

The selection of fine-grained versus coarse and probabilistic versus deterministic models of human performance has ramifications for system architecture and implementation. Specifically, data requirements, temporal representation, planning mechanisms, and knowledge representation will be affected. We will discuss our resolution of specific design variations as we describe the individual models in the systems.

In general, however, we have provided models of performance at resolutions adequate to describe the observable effects of operator action. For example, visual scanning behavior for radar screen search determines what an operator sees. Timing and position accuracy in describing human visual processes are critical and are modeled at a very fine level of response detail (positional variations of 1 degree of visual angle and temporal variations of 200 milliseconds are calculated). Alternatively, human decision processes are described in rule-based models that do not attempt to calculate decision times, as the effect of the decision is not critically dependent on small temporal variations. The architecture for system implementation is unique in that it allows models at varying levels of resolution to be used interactively (Elkind et al., 1989.)

With regard to probabilistic versus deterministic model assumptions, we decided, in conjunction with AFHRL, to restrict ourselves, in the present implementation, to deterministic models. This decision allows us to rule out probabilistic causes for automation impacts as revealed by the methodology. At finer detail, it also allows the effects of "miniscule" changes to be examined without significant confounding. If it is determined that probabilistic models of performance are required to capture critical performance, the architecture supports the use of these models in a multiple-run or Monte Carlo method of operation.

Model Integration

As described previously (Corker et al., 1989), our evaluation methodology is based on a modularized object-oriented paradigm for system representation. Human performance models used in this evaluation will be structured using this approach.[1] Models describing individual perceptual, motor, and cognitive processes are encoded as objects and methods on those objects. Communication among models (representing the process of perception, cognition, and action) is provided through LISP-based message-passing protocols. The action of these models is the sole basis for operator response to simulation. There is no higher-level repository of knowledge or provision of process.

Model Selection

We will describe here the currently implemented models that form the basis of human performance in the MCE. Note, however, that although these models represent our best integrated performance description to date, they should not be considered definitive or exclusive. The methodology has been structured to be robust in response to modification, removal, or replacement of any particular model.

---

1 There is no consensus among practitioners as to the "correct" integration approach. (See Chubb, Laughery, & Pritsker, 1987 for a discussion.) The rationale for our approach is provided in Corker et al. (1989).

MCE Operator Objects

Each of the operators modeled in the C^2 evaluation is an active and independent agent. Human operators take action based on their representation of the world.

World Representation. The operator object interacts with the MCE through perceptual processes and activities. The operator has an individually defined "updatable world representation." This world representation is a description of the world as the operator knows it. It contains rules for decision, briefing information, and an awareness of external events as they are passed through the operator's perceptual processes. The operator object assumes that information will be provided vocally and "heard," or presented visually and "seen." As discussed, all such information transactions in the object-oriented simulation take place through message-passing protocols among objects.

A human representation of the world is a complex structure the characterization of which is the topic of intense research efforts by experimental and cognitive psychologists. (See Collins and Smith, 1988, for a review of these issues.) A fundamental distinction is made in this representation based on whether knowledge about the world is stored as facts (termed declarative knowledge), or as actions and relationships (termed procedural knowledge). The objects in the simulation world are represented in a frame-theoretic paradigm. Objects are defined by characteristics called "slots," and those slots are filled by values supplied through the simulation. For instance, an aircraft-object in the simulation is defined by having altitude, velocity, bearing, expendables, etc. The operator's updatable world representation similarly represents the aircraft (once it is "seen" through the action of the visual perception mechanism) as an object with slot values that correspond to those of the original object. The world representation of objects, therefore, is fundamentally declarative. The internal state mirrors what is perceptually available from the external world, with two exceptions.

Those exceptions have to do with identification of a source of information and a temporal tag as to when the information is received. The source slot identifies from which piece of equipment, from what auditory source, or from what intelligence the information entered into the operator's internal world representation was derived. The temporal tag indicates when, in simulation time, the information entered the operator's world

representation. This tag is used to anticipate the spawning of required action. For example, the WD should check an aircraft's fuel status "X Ticks" after receiving word that the aircraft has been scrambled.

Perceptual, Cognitive, and Performance Models. The operator's world representation is composed of the initial state of the operator's knowledge, defined by the analyst. Changes to that representation occur as the operator-object interacts with the MCE simulation. That interaction takes place through vision, audition, memory, and motor responses. These processes are provided by LISP methods that act to support model function.

Visual Processing. The operators of MCE equipment must constantly "scan" their equipment to keep their mental image of the radar "Air Picture" information updated. In order to account for the time and movement required to find and fix target data in the MCE operator console visual field, we have implemented a model of visual scanning. Each of the activities in the mission simulation script has an attribute which identifies what equipment (and what sequence of interaction) is required to respond to mission demands. The majority of visual attention in MCE operation is required to be foveal; e.g., reading data, making bearing and range estimates, locating and operating the control panel switches. Foveal vision covers only a small part of the entire visual field. The region defined as foveal is .5 degree of visual angle, whereas peripheral vision approaches 180 degrees of visual angle (Graham, 1965). There will be two sorts of visual scanning that inform the internal representation of the operators according to the following mechanism.

The first "active gaze" represents the focused and directed movement from the current point of regard to a target point. The action is characteristic of actions in which the to-be-attended object is in a known position. The motion is a straight line from the present position to the target.[2] The operator is assumed to be 18 inches from the center of the Operator Control Unit (OCU). The velocity of ocular motion is 100 degrees per second. There is a 200-millisecond pause between eye motions (i.e., saccades). The visual scan will cover all of the displays of the OCU. The specific parameters that describe this model's operation (e.g., the distance of the operator from the screen, speed of ocular motion, and the dwell or

---

2 Though there may be a contribution to motion through head movement, we will not consider that at this time.

pause time) are variable slots in the model's definition. In this case, and in all other model definitions, we have attempted to instantiate the best data available regarding human performance to guide model operation. However, in every case, we have made the variables that define model function manipulable by the analyst to facilitate exploration of alternative functionality.

The second type of gaze is a monitoring or search pattern. The saccades in such a search pattern typically last for 50 milliseconds and cover about 10 degrees. Again, there is a 200-millisecond pause between movements. The effective radius of a fixation in this scan is about 14 degrees from the center of fixation (Bahill & Stark, 1979; Vossius & Young, 1962).

In addition to this basic distinction, gaze is directed by the decision-making and problem-solving tasks of the simulated operator. We have designed, and are implementing, the following visual dynamics into the simulation.

Visual Attendance

1. When a "viewable" referent is named (heard or spoken), thought about, or otherwise entered into a human being's attention, he or she tends to fixate the referent immediately. This tendency is more or less independent of whether or not the person seeks or requires information from the referent. However, when information-seeking or interpretation is not involved, such fixations may be brief.

2. The simulated operator will look at the referent to which it is attending. For example: When listening to a communication about a particular plane, the operator will shift its gaze to that aircraft. When thinking about the need to call a pilot about one thing or another, the operator will fixate the image of the relevant aircraft. When describing a pairing to a pilot, the operator will look at the image of the plane with which it is communicating, the bogie about which it is communicating, and the planned intersection point that it is communicating. These fixations will be coordinated with the verbal mention of the referents, thus constraining the speed of running off the whole pattern (Carpenter & Just, 1976; Cooper, 1974; Kahneman, 1973).

3. When no visual object relevant to the issue in attention is available for fixation (or when the available referent is displaying distracting characteristics), human viewers tend to direct their visual gaze to some non-informative and thus non-interfering locus, such as some empty point in the air between them and the screen (or any other visible surface), as long as they are concentrating on the related thought.

4. When the simulated operator has nothing to do and the screen has been relatively static, that operator will look at any new object that appears on the screen or that is moving or blinking. Note that this may be a reasonable heuristic for initiating operator attention to a developing scenario. In simulating the operator's mind and memory, such observed objects will be registered so that when the commander mentions them, they (and any other obvious or inferable characteristics) will already be represented in location in the operator's mind.

Spatial Problem Solving

5. Eye movements provide insight into the process of -- and, moreover, tend to mediate -- spatial problem-solving.

6. When the simulated operator is thinking about interception points and optimal pairings, its eye movements should mimic its thoughts. For *each* pairing considered, the operator's gaze will fixate on the target plane, the candidate friendly aircraft, and the projected point of intersection between them. If the operator must consider more than one pairing at a time in order to allocate friendlies to bogies properly, fixations on all such triads of locations will be included in the decision epoch. When the operator has settled on a pairing or set of pairings, the components of the pairing(s) should be fixated again to reflect and mentally record the decision.

7. When people view a moving object, they tend to compute the object's projected path and to use it in subsequent visual search for the object. The research literature does not permit parametric estimates of the robustness of this ability across time or intervening cognitive events. However, this ability can be expected to degrade in several different ways: (a) The greater the elapsed time since last attending to a moving object, the greater will be the x-y-z error in estimated location that results from imperfect estimates of the object's velocity; (b) Variance in estimated time since last attending to a moving object can only increase with

the amount of time that has elapsed; (c) The greater the number of intervening events since last attending to an object, the more difficult it will be for the operator to recall what he or she last knew about it (Carpenter & Just, 1976; Gould, 1976; Russo & Rosen, 1975).

8. If the simulated operator must return its gaze to a moving object (or if it must reestablish its location after killing a jammer), it will generally begin its search at the location where the object is projected to be rather than at the location at which the object was last attended.

Cognitive Processes

9. The cognitive requirements of MCE operation are extensive. They are also the least readily automated (see Table 1). First, there are issues of personal resource management: The operator has limited capacities in visual, auditory, cognitive, and psychomotor dimensions. We reasoned about these capacities and attempted to model task management. Each activity has associated with it a subjectively determined task loading. That load is estimated in terms of the amount of capacity (in visual, auditory, cognitive and psychomotor resources) that an operator needs to bring to bear to successfully complete a task. We then assumed that an operator will perform as many tasks as possible concurrently. That is, the operator will attempt to optimally schedule his/her activities.

Similarly, the operator has a limited capacity to remember the details of his or her activity in the MCE. These memory limits are currently implemented as a limited-length queue of interrupted or pending activities. Finally, the operator-objects must make reasoned decisions on action as the scenario evolves. This process is provided by rule-based decision mechanisms. These processes and their supporting models are detailed in a previous report (Corker et al., 1989).

Recently, developments have been made to these basic models. Specifically, a linear weighting algorithm has been implemented to augment pairing decisions (Henry, E. H., 1989). That algorithm considers factors such as friendly aircraft and bogie heading, speed, intercept points. controllers, areas of responsibility, and threat in determining a "value for pairing." The final specification of this algorithm is still under development. Memory model development includes concern for information value and refresh rates in determining what

and when items will be forgotten. Finally, rule-based decisions are being augmented to consider higher-level abstractions in rule application.

As the system develops to include concern for battle management, and situation-dependent decision-making, more sophisticated reasoning processes will need to be represented; e.g., evidential reasoning and situation assessment (Adams & Pew, 1989; Lowrence, Garvey, & Strat, 1986). In addition, attention mechanisms will need to be provided to guide operator behavior from a problem-solving perspective.

IV. SYSTEM ARCHITECTURE

The C^2 evaluation methodology is implemented on a Symbolics 3670 Computer Genera 7.2™ operating and color system. The system requires approximately 2 megabytes of memory for loading and operation.

The basic architecture for the system is illustrated in Figure 2. The system is implemented in a modular and object-oriented framework, as discussed in the Software User's Manual (Corker, 1989b) and the Operational Concept Document (Corker, 1989a). The module boundaries in Figure 2 correspond to the conceptual design and implementation distinctions in the system.

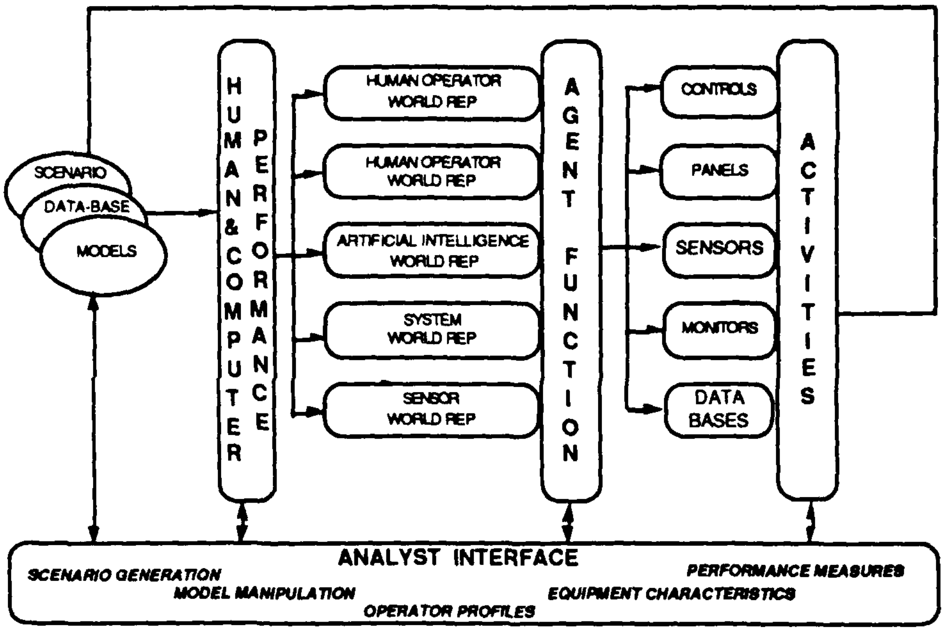

Figure 2. Basic Architecture for Human/System Performance Analysis Simulation.

The functional flow through the implementation begins with models of the scenario and environment in which the evaluation methodology is to be exercised. In the case of the MCE automation impact evaluation, this scenario includes geographic and geopolitical boundaries in support of an air defense operation. Included in this description are object-based representations of Friendly and Enemy aircraft, radar sites, CAP points, and airbases. The activity of the scenario is played out though the emulation of MCE equipment in the MCE operator console. (This equipment is illustrated in the equipment description ovals attached to activities.) The scenario is interpreted through the human operator performance models (described in the previous section). The output of these models provides data that modify the world representation of the operators that have interacted with the displays.

The agent function module is composed of operator-independent abstractions of the processes by which the operator-objects act on the data contained in their world representation. These abstractions currently include communication protocols, interruption/resumption protocols, task-queue management operations, and decision mechanisms. Once action is decided upon, the way in which that action takes place is mediated by descriptions of the system equipment. In the current instantiation, that equipment is the MCE OCU.

Finally, activities are initiated which describe when, how, for how long, and with what resources the operator responds. The effect of these activities is then fed back in order to reflect changes in the scenario state as a result of operator action.

For example, as the MCE RGDU displays the appearance of aircraft (directed by the scenario script), the human visual performance models direct the position and dwell time of the WAO's gaze as the WAO searches over the MCE OCU. The information (MCE object status) that is taken in by the WAO is used to update his/her world representation. The state of objects in the world representation is arranged by categories, and attached to these categories are rules of behavior for the WAO under the current rules of engagement and air tasking order (ATO). To continue the example: If the WAO's visual scan encounters a Friendly symbol as it passes over the RGDU, a set of rules attached to Friendly aircraft is run to see if the condition of that aircraft meets the criteria for any rules to be activated, or "fired."

Attached to Friendly aircraft objects are rules regarding their combat mission state (e.g., paired, enroute, engaged, on CAP) and action to be taken by the WAO based on time since last observation. These rules can either cause action to be initiated or simply cause the WAO's world representation to be updated. The WAO's rules generally dictate that, given a particular condition of the air battle, communications be initiated either within the MCE (e.g., to direct a WD to communicate with a pilot or alter the current pairings and assignments) or through communication with an external agency (e.g., to scramble fighters to CAP or to an engagement). The firing of the appropriate rule for action causes an activity to be created (spawned). In turn, that action may have several supporting actions that must be taken in order to satisfy the termination conditions for that state of action. An activity (e.g., initiate communications) invokes models that describe procedural sequences (what must be done), communications protocols (how it must be done), and motor response requirements (what are the physical parameters for its completion).

Other human operator agents within the MCE are guided by similar perceptual models, but the rules and the activities spawned by those rules depend on the duties and the profile of that operator. So, to continue the example above: A call from the WAO to a WD results in an auditory input to the WD. The WD responds to this change in world state by applying rules to the content of the communication that spawn activities on his/her part. For example, a request from the WAO to re-pair a fighter will result in the requested re-pairing, and in a new condition (a previously paired fighter now unpaired). The WD object must

determine, according to mission state and rules of engagement, what is to be done with that fighter. Rules for pairing geometry are invoked which spawn action. Finally, action is effected through the procedures required by the MCE equipment suite.

In addition to these modules, the system provides a set of interface tools designed to facilitate screen-based, mouse-activated manipulation of the objects that comprise the evaluation system scenario.

Function Flow

The basic architecture for LISP functional flow for the MCE is illustrated in Figure 3. It is presented there from the point-of-view of the management of information display. The simulation module is essentially the forcing function for the flow of activity in the analysis. Events are scheduled to occur at particular simulation times (e.g., a particular simulation "Tick"), or as an analyst-selected "Asynchronous Event" that is invoked at any time with a screen-based command. It is important to note that the simulation module serves as a stimulus to the operator models.

The major control modules are a Master Controller and Event Handler. The next level of control is found in the operation of the Model Displays, MCE Display, Radar (RGDU) Display, and Agent Top-Level object modules. Below these are the individual models, the actions of the simulation agents, the function of the MCE equipment, and activities of the radar screen. We will describe the operation of these modules in some detail.

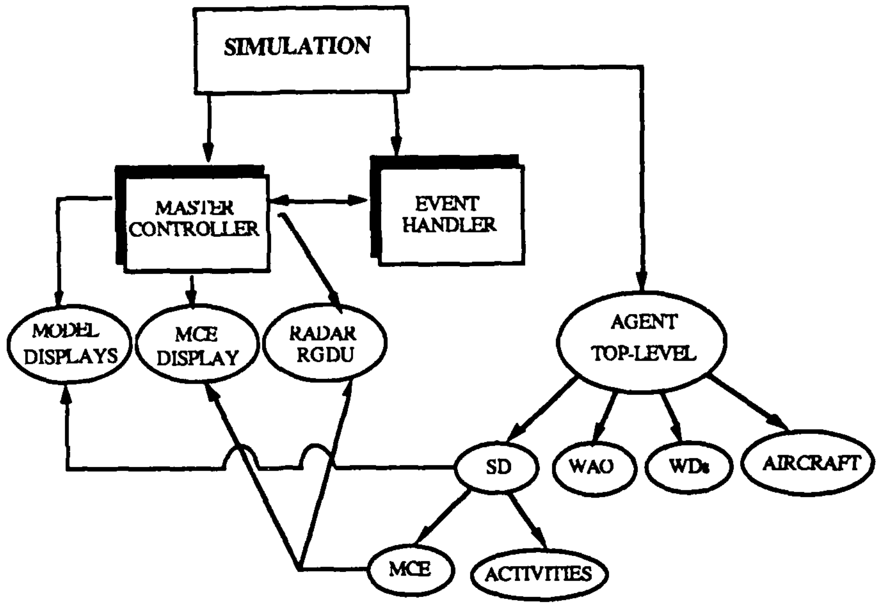

Figure 3. **Functional flow for the MCE.**

The Master Controller acts as the system executive and routes information and messages among the system modules. The Master is linked to the Event Handler, which contains two types of events that drive the operation of the simulation objects. "Tick-based" events are the basic script of the simulation and depend on the initial configuration of the agents including Bogie/Friendly aircraft and the rules of engagement. The other event type is "asynchronous," which provides for the injection of user-defined events into the operation of the simulation. This also provides a mechanism for conditional events being defined in the simulation. Event streams through the Master drive the displays, the radar, and the activities of the agents.

In addition to the display modules for models and equipment, each agent contains an object pre-presentation of the MCE. The radar is represented only in the radar display object and is referred to by the agents through the Master.

Display Controller

The primary means by which the C[2] evaluation system manipulates screen displays is through the use of a Display Handler. A given subsytem in the C[2] evaluation methodology system can have several displays that it must show. Here the term "display" is used to mean

any self-contained image (a table, graph, button-grid, radar-screen, text output, etc.) that the system can output to the screen. A display typically has a dedicated output window, although this is not required.

Assigned to each display is an object called a "Display Handler." As its name implies, this Display Handler is responsible for showing its corresponding display on the screen. The Display Handler governs all aspects of the display, including simply drawing the display, updating the output as the values of the program using it change, and refreshing the present state of the display in response to specific refresh commands or as the display appears or reappears on the screen.

Furthermore, any information about mouse-clicks or mouse-motions that occur on the window is transferred to the Display Handler currently governing that window. (If no Display Handler is assigned to that window, mouse-clicks, etc. are ignored.) The Display Handler is then free to respond to such clicks or motions in a way that is appropriate for its display.

Similarly, all keyboard input is passed along to the system itself, which is responsible for routing the input data to the Handler for the display that it currently has selected to receive the input. The Handler then uses the data as appropriate; for example, as commands to the system or as system input (e.g., as data entry in a table).

A Display Handler processes its graphics commands by acting on an object called its "PWindow" (i.e., a Pseudo-Window). This object contains all the information that is specific to the windowing system on the underlying hardware platform system in use. The PWindow is responsible for translating any of the standard set of graphics messages that it can receive from its parent Display Handler into a command format appropriate for the hardware-platform-specific windows on which the application is currently running.

Display Handler Features

In keeping with our design goal of system modularity, the Display Handler paradigm provides that the physical-platform-specific window is maintained by a simple, passive display device. The window has no application-specific role. The window does not reference any of the operational details of the application whose display it contains. This

application-independent operation has two exceptions: the transferal of mouse-click information and the notification of keyboard entries. In these cases, the window must pass information about what type of mouse-click has occurred, and where or what keyboard entries were made.

More precisely, all that is actually required of a platform-specific window is:

1. That it can handle a standard set of output commands (e.g., a graphics command like DRAW-LINE/CIRCLE/RECTANGLE), and string-output commands, and

2. That it is capable of "remembering" and keeping track of which Display Handler (if any) is currently doing output on it, so that

3. It can transmit to its corresponding Display Handler data about the mouse-clicks (and possibly moves) that it receives, and

4. It can pass along keyboard information by, for example, placing any characters it receives into a common input queue.

Example of Use of the Display Handler Paradigm

As a specific example, we will consider the C^2 evaluation methodology's models of the workload and performance of a crew operating the MCE system. There are two major clusters of displays in the system. These are described in detail in the Software User's Manual (Corker, 1989b) and Operational Concept Document (Corker, 1989a).

First, a full-color representation of the console of the MCE system is shown. This display has two parts: a radar screen and a complicated set of pushbuttons and button-related panels. On the radar screen are a number of display icons representing the controlled aircraft and the symbology assigned to the aircraft by the MCE system. Associated with each aircraft is a Display Handler, which is responsible for showing the aircraft's symbology, its radar return, various textual displays associated with the aircraft, highlighting/markings that the MCE system can associate with the aircraft, etc. A Display Handler is also associated with each grid of buttons on the MCE console display. Each Display Handler is responsible for showing the buttons in the Display Handler's corresponding grid, displaying the button in each of its various possible pushed-states, accepting mouse-clicks on the button, etc.

Second, a monochrome display set shows the current workload and state of the human crewmember models. Each crewmember model has two displays associated with it:

1. An animated "rolling paper-tape"-like display that shows a set of four bar charts displaying the time-dependence of the workload on the crewmember, and

2. A textual display showing the current configuration of the rules governing the behavior of the crewmember according to the system's current internal model.

Each crewmember model has a Display Handler governing its output to each of these two displays. In addition to a pair of displays for each of the crewmembers, there is a corresponding pair of display windows for the airbases, aircraft, and TACC (supreme command) used by the simulation. A similar pair of windows is used for system output and display. Each crewmember model's displays are fully independent, so that the analyst/user running the system is able to select from among the various crewmember models that set which models the information it wishes to display.

This model of window output/interactions has a number of significant advantages. True portability is enhanced. Because of the inherently hardware-specific nature of windowing systems, displays and window interactions can be the most difficult features of an existing application to port to a new hardware platform. However, as noted above, in the Display Handler model all knowledge about the nature of and interactions with the platform-specific windows being used by the displays is highly localized and made modular by being encapsulated within the PWindow. As a result, in porting the display-related portions of an applications system to a new hardware platform the only portion that needs to be modified is the PWindow itself. Moreover, this conversion is a one-time cost; specifically, it need not be done on a per-application basis. Once a platform-specific version of a PWindow has been established, it can be reused for future ports of Display-Handler-based systems to that hardware platform.

All windows in a given system are completely interchangeable. Again, no knowledge about how the display is to be shown is embedded in the platform-specific window. Consequently, rearranging or redistributing displays for a given system is simply a matter of

reassigning the physical windows among the appropriate Display Handlers and their PWindows.

Furthermore, this greatly simplifies the resourcing of these windows. Rarely used or complicated types of displays need not have their possibly space-expensive windows created for a single, short-term use. In other words, there need be no more windows created than the maximal number that can be shown at a single time, regardless of their use. In fact, it is simple to turn off any or all displays in a given system.

As an example, in the C^2 evaluation system discussed above, the user/analyst can select which set of crewmember model output he or she wishes to display. In this model of window interactions, showing the chosen displays becomes simply a matter of distributing the necessary windows among the Display Handlers for the models whose output is desired.

Having all output to the screen channeled through the various Display Handlers provides the system with a centralized locus for controlling, manipulating, or eliminating some or all of its output. Indeed, a program outputting through a specific display need not even know whether its output is actually being shown. This has two advantages:

1. A given portion of a system need not be concerned about whether it is doing output (as stated above, if a disabled display is later re-enabled, the Display Handler is responsible for updating the display appropriately). In the C^2 evaluation displays of crewmember model data, all output from a given crewmember model is passed through a single Display Handler. This gives the system a useful, simple way for turning off the display from that model, and no model is concerned as to whether its output is actually being shown.

2. In certain applications (e.g., complicated, graphics-intensive simulations) where a significant portion of run-time often is devoted to graphics and textual output, the Display Handler model gives a single, central location for disabling all output to a display when this is desired. In the C^2 evaluation system, with its many complicated displays, it is often desirable, when attempting to run until a specific predetermined time-step, to disable all displays until the run is over. Suppressing the displays for these intermediate steps can allow a great improvement in speed.

Multiple Highlighting is Simplified.

Highlighting for emphasis is a generally desirable feature of a system. Multiple highlighting, such that when a single entity in the system is somehow selected or singled out to be noticed, all representations of that entity currently on the screen become highlighted, is also useful.

Under the Display Handler paradigm, highlighting is simply another aspect of the details of a particular Display handled by a Display Handler. When an entity in the system is told that it needs to highlight its representations on the display screen, it notifies the Display Handlers responsible for the displays in which the entity occurs.

Agents

Contained in each crewmember agent is a model of that crewmember's behavior (encapsulated in his/her current set of activities) and his/her MCE console. Associated with each crewmember's activity state is a set of displays. Figure 4 shows the agent structure.

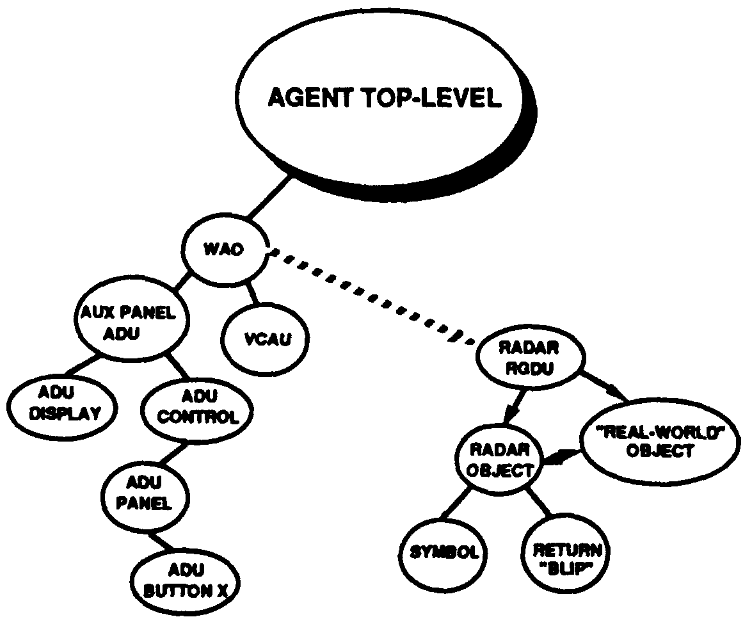

Figure 4. Agent Structure.

First, there is a strip-chart display which shows a continually updated representation of the Visual, Auditory, Cognitive, and Psychomotor (VACP) load on the crewmember over time. Second, there is a textual representation of the hierarchy of activities that the agent is currently running. Both of these displays are shown on the monochrome screen. On the color screen is a display showing the current and last communications from and to the agent.

Associated with each of these displays is a Display Controller object which governs and controls how each of the displays is shown. This Controller is responsible for knowing, among other things, whether the display is currently disabled and how much room it has on the screen.

Each crewmember agent also contains an internal representation of his/her own MCE console. Internally, the states of all the buttons, etc. are recorded and maintained. A view of only one crewmember agent's MCE is shown at any one time. At that time, the MCE console displays of all the other agents are disabled, without affecting in any way the internal representation of the MCE's state (for more on this point, see the discussion of the Display Controllers). The sole exception to this is the representation of the radar and its display; at present, a single radar object is held in common and used by all of the agents' internal representations of their MCE consoles.

An agent in the C^2 system is used to represent an independent, free-standing entity capable of and responsible for initiating its own behavior. This behavior is controlled by the set of activities that the agent is currently running. These activities are spawned in response to changes in the agent's environment in the system.

An activity is a unit of behavior governing the action of an agent. Roughly, it is a piece of code that runs for a short duration, until a given goal is accomplished or until it is otherwise terminated. During its lifetime it will, at various times, execute code to affect the behavior of its parent agent or send messages to other agents in the system or features of the simulation.

The structure of an activity can be recursively hierarchical; that is, it can itself spawn "children" subactivities in order to delegate subtasks that need to be performed. For example, a crewmember agent might need to communicate with another crewmember, as a

response to the appearance of an unknown object on his/her radar screen. To accomplish this task, the crewmember agent would spawn a high-level "communicate with crewmember" activity. This activity would have small subactivities such as "dialing" up the other crewmember, talking to the other crewmember, and "hanging up" the communication. Each of these tasks would, in turn, have many subtasks (reaching for and pushing buttons, looking to verify that a button-click "took,"etc.).

An example of a complicated, high-level agent is that of the crewmembers in the C^2 system. These agents, as models of human behavior, gain information about their environment -- as described by the system -- by means of their auditory and visual models. The human model/agents then respond to the changes in their resultant internal model of the world by modifying the set of activities that the agents have running at that moment.

The internal representation of the world of these agents is governed by their auditory and visual models. At given intervals, the agent looks (or listens) to its environment and collects information about the world, which it stores in its memory. According to what it then perceives about the world, it responds to the world by modifying the set of activities it is running.

An example of a simpler type of agent is aircraft objects in the C^2 system. These objects also respond to changes in their environment by spawning new activities, but the model of their interactions with the rest of the world is much simpler. In short, they simply respond to incoming messages sent to them by the human crewmember agents. (For example, the crewmember agent WD1 might send an aircraft agent a command to return to an airbase for refueling.)

The channels of world interaction for these agents are much simpler than they are for the human agent. Basically these agents merely receive and respond to direct communications from the crewmember agents.

Governing all the agents is an entity known as the Agent Top-Level. The Agent Top-Level is responsible for keeping track of and handling communications among the various agent objects and the other entities of the system. It is also responsible for various system maintenance "housekeeping" tasks such as making sure all the agents are "ticked" at the appropriate moment.

Activities

Activities are the structures that make up the procedures an operator performs in response to the simulation events and according to his/her responsibilities and rules. Activities are encoded as LISP procedures and methods that describe what is to be done, what are the enabling conditions for that performance, who takes this action, the action's duration and load, how the action is successfully completed, and how that Activity is terminated or interrupted.

Each tick-step for a given agent is divided into three parts or passes:

1. Pre-Tick. In this pass, the set of Activities that the agent will run for the specific tick is decided on. The Agent is asked which of its current set of Activities it will run on this tick. This decision can be very simple. For instance, in the aircraft object Agents, all the available Activities are in a strict linear order of precedence, and the currently available Activity with the highest priority gets to run.

Alternatively, for the human Agents that depend on the VACP load models, each agent must first sort his/her current set of Activities according to a preset priority. Next, this set is then gone through in order and each top-level or parent activity is asked to decide if it "wants" to run on this tick; that is, are the conditions appropriate for it to run on this tick? If not, this Activity is skipped over. Finally, if the Activity can be run, its VACP for this tick is calculated and the corresponding total loads for the Agent are incremented. If one of the four V, A, C or P loads becomes too great, this Activity cannot be run this tick. This process continues until all the Activities are processed or all the Loads are filled.

2. Tick. In this pass, the actual work of the Activity is done; messages are sent to other entities, etc.

3. Post-Tick. In this pass, some side-effects of the tick are cleaned up. For instance, if the activity spawned a new activity during the Tick pass, it is actually queued-up and spawned during the the Post-Tick phase. (This is done to avoid a cascade effect, in which an Agent that receives its tick after the current Agent would effectively get a resulting message one tick out of phase with the current Agent.)

For example, in the Activities governing communications, the Agent who has initiated the communication has a "send communication" Activity, and the Agent to whom the communication was sent has a "receive communication" Activity. The structure of individual communication is illustrated in Figure 5. The figure depicts typical communication between the WAO Agent and the WD1 Agent. In this case, the WAO is the initiator of communication and the WD1 is the recipient of that communication.

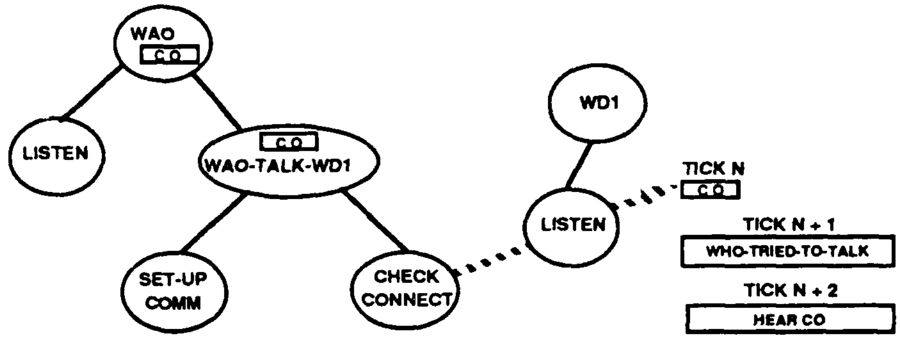

Figure 5. Communication

During the Pre-Tick pass for sending, the Agent, after first determining that no activity of higher priority is pending, must decide if it is still appropriate for the "Send Communication" Activity to run. For example, the sending Agent checks to see if the receiving Agent is still connected, or if the receiver has been interrupted by activities of his/her own with higher priority than listening to the communication. If it is still appropriate, the message is sent. During the Tick pass, the receiving Agent determines who is trying to communicate and whether that communication is of sufficiently high priority to be heard. In the Post-Tick pass, the receiving Agent actually hears the content of the message.

The simulation issue addressed in this multi-pass paradigm is that Activities or communications on the part of one Agent may change the world situation and context of action on the part of another Agent in the team. Queueing and prioritization are required, as well as a period in which to allow decisions to settle into the new context which each "tick" of Activities brings to the situation.

V. CONCLUSIONS

The present report has described the implementation of an evaluation methodology to assess the impact of automation on the performance of C^2 systems. The system was implemented using an MCE-equipped Control and Reporting Center as the test bed for the methodology; however, the conceptual design and software implementation of the system can support general application across a broad spectrum of man/system interactions. This generality of application comes from an extensive design effort in the system's software architecture. In addition to internal modularity and strict adherence to an object-oriented programming paradigm, the system design focused on maintaining independent representations for such global notions as "Mission," "Equipment," and "Operators." The independence of these concepts provides to an analyst the ability to make changes in the assumptions, requirements and characteristics of the components of the C^2 system. The methodology is also "executable"; that is, the effect of changes in the C^2 system objects can be examined by running the system in a simulation mode.

The system is designed to operate in two modes. In an analytic simulation mode, the human operators of the system are represented in software through the interaction of a number of performance models. There is a continuing requirement to refine and extend these performance models to include memory-directed activity, attention-sensitive performance, and improved representation of perceptual processes. In a manned-simulation mode of operation, one or more of the software operators is replaced with real human operators. This development has led to research concerns for the successful interface of a human operator into a simulation through voice recognition, speech synthesis and touch panel control. In addition, there are issues regarding generality, timing and stability of such human-in-the-loop operation.

VI. REFERENCES

Adams, M. J., & Pew, R. (1989). Cognitive Management of Complex Semi-Automated Systems. Proceeding of the IEEE International Conference on Systems, Man, and Cybernetics, (pp. 1216-1220). Cambridge, MA.

Bahill, A. T., & Stark, L. (1979). The trajectories of saccadic eye movements. Scientific American, 240 (pp. 108-117).

Carpenter, P. A., & Just, M. A. (1976). Linguistic influences in picture scanning. In R. A. Monty & J. W. Senders (Eds.), Eye movements and psychological processes. Hillsdale, NJ: Erlbaum.

Chubb, G. P., Laughery, K. R., & Pritsker, A. B. (1988). Simulating manned systems. In G. Salvendy (Ed.), Handbook of human factors (pp. 1298-1327). New York: John Wiley & Sons.

Collins, A., & Smith, E. (1988). Readings in cognitive science. San Mateo: Morgan Kaufmann Publishers.

Cooper, R. M. (1974). The control of eye fixation by meaning of spoken language. A new methodology for the real-time investigation of speech perception, memory and language processing. Cognitive Psychology, 6, (pp. 84-107).

Corker, K. M. (1989a). System operational concept document. (Unpublished manuscript). Cambridge, MA.

Corker, K. M. (1989b). Software user's manual. (Unpublished manuscript). Cambridge, MA.

Corker, K. M., Cramer, N. L., Henry, E. H., & Spaeth, R. L. (1989). Methodology for Evaluation of Automation Impacts on Tactical Command Control C^2 Systems: Domain Selection and Approach (AFHRL-TR-89-17), Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

Elkind, J. I., Card, S., Hochberg, J., & Huey, B. (1989). Human performance models for computer-aided engineering. Washington, DC: National Academy Press.

Gould, J. D. (1976). Looking at pictures. In R. A. Monty & J. W. Senders (Eds.), Eye movements and psychological processes. Hillsdale, NJ: Erlbaum.

Graham, C. (Ed.) (1965). Vision and visual perception. New York: John Wiley & Sons.

Henry, E. H. (1989). Personal communication. Air Force Human Resources Laboratory, Wright-Patterson Air Force Base, OH.

Kahneman, D. (1973). Attention and effort. Englewood Cliffs, NJ: Prentice-Hall.

Lowrence, J., Garvey, T. & Strat, T. (1986). A Framework for Evidential-Reasoning Systems. Proceedings of the 5th AAAI (pp. 896-903). Philadelphia, PA.

Rouse, W. B. & Boff, K. R. (1987). System Design, North Holland, NY.

Russo, J. E., & Rosen, L. D. (1975). An eye fixation analysis of multialternative choice. Memory and Cognition, 3 (pp. 267-276).

Salvendy, G. (Ed.) (1988). Handbook of human factors. New York: John Wiley & Sons.

Stark, L., Vossius, G., & Young, L. (1962). Predictive control of eye-tracking movements. Transactions of Human Factors Electronics, HFE-3 (pp. 52-57).

Woods, D., Roth, E., Hanes, L. F., & Embrey, D. (1986). Models of cognitive behavior in nuclear power plant personnel: A feasibility study. (Report #NUREG-CR-4532) U.S. Nuclear Regulatory Commission

VII. GLOSSARY

ADU	Auxiliary Display Unit
AFMRL	Air Force Medical Research Laboratory
AFHRL	Air Force Human Resources Laboratory
AI	Air Intelligence
AOR	Area of Responsibility
ASOC	Air Support Operations Center
ATO	Air Tasking Order
BAI	Battlefield Air Interdiction
C^2	Command and Control
CAP	Combat Air Patrol
CAS	Close Air Support
CRC	Control and Reporting Center
CRP	Control and Reporting Post
CRT	Cathode-Ray Tube
DCA	Defensive Counter Air
FACP	Forward Air Control Post
FEBA	Forward Edge of the Battle-Area
FOG	Finger-on-Glass
GACC	Ground Attack Control Capability
GCI	Ground-Controlled Intercept
MCE	Modular Control Equipment
OCA	Offensive Counter Air
OCM	Operation Control Modules
OCS	Operator Control Stations
OCU	Operator Control Unit
OCU	Operator Control Units
OM	Operations Modules
PWindow	Pseudo Window
RGDU	Radar Graphics Display Unit
TACC	Tactical Air Control Center
TACS	Tactical Air Control System
USAF	United States Air Force
VACP	Visual, Auditory, Cognitive Psychomotor
VCAU	Voice Communications Access Unit
VID	Visually Identification
WAO	Weapons Assignment Officer
WD	Weapons Director

* U. S. GOVERNMENT PRINTING OFFICE: 1990--761-051/20064